UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,904	03/31/2004	Eric F. Vannerson	42P19126	9294

8791 7590 10/19/2007

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	**PAPER NUMBER**
2183	

MAIL DATE	DELIVERY MODE
10/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/815,904	VANNERSON ET AL.
	Examiner	Art Unit
	David J. Huisman	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 August 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3 and 5-29* is/are pending in the application.

4a) Of the above claim(s) *10-17* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-9 and 18-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 August 2007* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-3, 5-9, and 18-29 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 8/13/2007.

Specification

3. The amended title of the invention, submitted by applicant on August 13, 2007, is still not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. It is suggested that applicant incorporate, into the title, the attaching of breakpoint bits to instructions and the manipulation of three fields added to a processor control status register.

4. As stated in the previous Office Action, the abstract of the disclosure is objected to because it does not sufficiently describe the disclosure and, therefore, does not assist readers in deciding whether there is a need for consulting the full patent text for details. The abstract merely includes a list of connections and not what the invention does. The examiner requests that a new, more detailed abstract be submitted. Correction is required. See MPEP § 608.01(b).

Drawings

5. The drawings are objected to because of the following minor informalities:

- In Fig.8B, in both portions, it is not clear what is meant by LDTI/LDFI into debug instruction. How is an instruction into another instruction? What does this mean?

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

6. Claim 24 is objected to because of the following informalities: If applicant wishes to use "adding" instead of --attaching-- in line 2, then please replace "attached" with --added-- in line 6 for consistency and increased clarity. Appropriate correction is required.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

8. Claims 1, 18, and 24 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Specifically, each of the aforementioned claims is unclear because the examiner does not know what is meant by adding at least three fields to a control status register. The examiner asserts that unless the processor is reconfigurable, which applicant's specification gives no indication of, a register is a fixed hardware component. Additional hardware cannot be dynamically attached to pre-existing hardware. Does applicant instead mean that at least three bits of a register are set? Fig.6 shows the three bits in question but the examiner is not sure how these are added without undue experimentation. The examiner can only assume that applicant has these three bit fields already existing at run-time (i.e., the system is fabricated in such a manner to include these three bits), and then applicant manipulates these bits. It is noted that the examiner finds no real difference in "attaching" and "adding". Either way, it is not clear how this is done, and the examiner has been unable to find applicant's explanation in the remarks which would overcome this rejection.

9. Claims 2-3, 5-9, 19-23, and 25-29 are rejected under 35 U.S.C 112, 1st paragraph, for being non-enabling, because they are dependent, either directly or indirectly, on a non-enabling claim.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1-3, 5-9, and 18-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glew et al., U.S. Patent No. 5,694,589 (herein referred to as Glew), in view of Deng et al., U.S. Patent No. 6,951,416 (herein referred to as Deng).

12. Referring to claim 1, Glew has taught an apparatus comprising:

a) a plurality of processors coupled to a controller and a memory. See Fig.1, Fig.3, components 100 (memory) and 200 (controller), and Fig.4. Note that a superscalar processor includes N processing elements to execute up to N instructions in parallel (this is the inherent nature of a superscalar system). Hence, the N processing elements are the plurality of processors, as they each process data.

b) the controller to execute a debug process, said debug process attaches at least one breakpoint bit field directly to each of a plurality of processor instructions. See Fig.3, at least component 200, Fig.4, and column 6, lines 24-38.

c) Glew has not taught that said controller adds at least three debug register bit fields to at least one processor control status register field, wherein said at least three register bit fields comprise a run field, a single step field and a debug enable field. However, Deng has taught such a concept. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a

single-step field, and field 202 (BE1, BE2, BE3, or BE4), which is a run field. These bits allow for increased functionality and flexibility in debugging. For instance, the enable/disable field allows debugging to be turned on and off, the single-step field allows a break after each instruction, which allows a user to view the effects of each instruction on the system, and the run field allows breakpoints to be enabled/disabled for specific addresses. So, even though an instruction should break because it corresponds to an address, the user may decide skip that break and allow the processor to continue running for whatever reason. As a result, in order to increase flexibility and functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the controller adds at least three debug register bit fields to at least one processor control status register, wherein said at least three register bit fields comprise a run field, a single step field and a debug enable field.

13. Referring to claim 2, Glew in view of Deng has taught the apparatus of claim 1, wherein said at least one breakpoint bit field allows a breakpoint to be one of set and not set for each of said plurality of instructions. See column 6, lines 24-38.

14. Referring to claim 3, Glew in view of Deng has taught the apparatus of claim 2, wherein a breakpoint bit set for an instruction is associated with the address of the instruction. See Fig.3, Fig.4, and column 6, lines 24-38. Note that the address of the instruction is used by the controller to attach the appropriate bit value.

15. Referring to claim 5, Glew in view of Deng has taught the apparatus of claim 1, wherein said single step field allows a set of instructions to each be single-stepped through one cycle at a time. See Fig.16, field 206, and column 5, lines 29-33, of Deng.

16. Referring to claim 6, Glew in view of Deng has taught the apparatus of claim 1, wherein said debug enable field one of enables and disables a debug mode. See Fig.16, field 208, of Deng.

17. Referring to claim 7, Glew in view of Deng has taught an apparatus as described in claim 1. Glew has not explicitly taught that at least one instruction loads content of at least one register into an instruction memory coupled to at least one processor of the plurality of processors via a bus. However, Official Notice is taken that store instructions and their advantages are well known and accepted in the art. A store instruction, as is known, stores data in a register into an instruction memory (a memory which is accessed by an instruction). Such an instruction allows for increasing the storage space of the program as an amount of data that is too large to store in the register file can also be stored in the memory and retrieved later when necessary. As a result, in order to increase storage flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew to include a store instruction that loads content of at least one register into an instruction memory coupled to said at least one processor via a bus.

18. Referring to claim 8, Glew in view of Deng has taught an apparatus as described in claim 7. Glew has not explicitly taught that content of said instruction memory is loaded into a register coupled to at least one processor of the plurality of processors. However, Official Notice is taken that load instructions and their advantages are well known and accepted in the art. A load instruction, as is known, retrieves data from instruction memory (a memory which is accessed by an instruction) and loads the data into a register. Such an instruction allows for increasing the storage space of the program as an amount of data that is too large to store in the register file can also be stored in the memory and retrieved later when necessary. As a result, in order to increase

storage flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew to include a load instruction that loads content of said instruction memory into a register coupled to said at least one processor.

19. Referring to claim 9, Glew in view of Deng has taught the apparatus of claim 1, wherein internal states of each of said plurality of processors are accessible through said debug process. The examiner deems this to be inherent as this is the nature of debugging. During debugging, a user is able to view the state of the machine (and consequently, of each of the processors) in order to determine how the system is functioning.

20. Referring to claim 18, Glew has taught an apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:

a) adding at least one breakpoint bit field directly to each of a plurality of instructions. See Fig.3, at least component 200, Fig.4, and column 6, lines 24-38.

b) Glew has not taught adding at least three debug register bit fields directly to at least one processor control status register field. However, Deng has taught such a concept. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and field 202 (BE1, BE2, BE3, or BE4), which is a run field. These bits allow for increased functionality and flexibility in debugging. For instance, the enable/disable field allows debugging to be turned on and off, the single-step field allows a break after each instruction, which allows a user to view the effects of each instruction on the system, and the run field allows breakpoints to be enabled/disabled for specific addresses. So, even though an instruction should break because it corresponds to an address, the user may decide skip that break and allow the

processor to continue running for whatever reason. As a result, in order to increase flexibility and functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that at least three debug register bit fields are attached to at least one control status register.

21. Referring to claim 19, Glew in view of Deng has taught the apparatus of claim 18, further containing instructions which, when executed by a machine, cause the machine to perform operations including: determining a state of said breakpoint bit, and setting a breakpoint for an instruction if it is determined that said state of said at least one breakpoint bit field is set. See Glew, column 6, lines 24-38.

22. Referring to claim 20, Glew in view of Deng has taught the apparatus of claim 18, wherein said at least three register bit fields comprise a run field, a single step field and a debug enable field. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and field 202 (BE1, BE2, BE3, or BE4), which is a run field.

23. Referring to claim 21, Glew in view of Deng has taught the apparatus of claim 20, further containing instructions which, when executed by a machine, cause the machine to perform operations including: determining a state of a run field bit, and running a set of instructions if said state of said run field bit is set, and stopping a set of instructions if said state of said run field bit is not set. See Deng, Fig.16, and note that if any first field of four-bit field 202 is enabled, then a set of instructions is stopped when the address associated with the first field. The instructions stopped are the instruction associated with the address and all subsequent instructions. If the first field is disabled, then the set of instructions will run because the address associated with the first field does not result in a breakpoint.

24. Referring to claim 22, Glew in view of Deng has taught the apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including: determining a state of a single step bit, single-stepping through a set of instructions for a cycle if said state of said single-step bit is set. See Fig.16, field 206, and column 5, lines 29-33, of Deng.

25. Referring to claim 23, Glew in view of Deng has taught an apparatus as described in claim 18.

a) Glew has further taught instructions which, when executed by a machine, cause the machine to perform operations including: accessing internal states of each of a plurality of processors through said debug process. The examiner deems this to be inherent as this is the nature of debugging. During debugging, a user is able to view the state of the machine (and consequently, of each of the processors (each superscalar processing element) in order to determine how the system is functioning.

b) Glew in view of Deng has not explicitly taught that the apparatus further contains instructions which, when executed by a machine, cause the machine to perform operations including: loading content of at least one register into an instruction memory and loading content of said instruction memory into the at least one register. However, Official Notice is taken that store and load instructions and their advantages are well known and accepted in the art. A store instruction, as is known, stores data in a register into an instruction memory (a memory which is accessed by an instruction), and a load instruction, as is known, retrieves data from instruction memory (a memory which is accessed by an instruction) and loads the data into a register. Such instructions allow a processor to read and write data to memory, thereby increasing its storage space. As a

result, in order to increase storage flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew in view of Deng to include a load content of at least one register into an instruction memory (store instruction) and load content of said instruction memory into the at least one register (load instruction).

26. Referring to claim 24, Glew has taught a method comprising:

a) adding at least one breakpoint bit field directly to each of a plurality of instructions, wherein the attached at least one breakpoint bit field is an additional field directly added to each processor instruction. See Fig.3, at least component 200, Fig.4, and column 6, lines 24-38.

b) Glew has not taught adding at least three breakpoint register bit fields to at least one processor control status register field. However, Deng has taught such a concept. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and field 202 (BE1, BE2, BE3, or BE4), which is a run field. These bits allow for increased functionality and flexibility in debugging. For instance, the enable/disable field allows debugging to be turned on and off, the single-step field allows a break after each instruction, which allows a user to view the effects of each instruction on the system, and the run field allows breakpoints to be enabled/disabled for specific addresses. So, even though an instruction should break because it corresponds to an address, the user may decide skip that break and allow the processor to continue running for whatever reason. As a result, in order to increase flexibility and functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that at least three debug register bit fields are attached to at least one control status register.

27. Referring to claim 25, Glew in view of Deng has taught the method of claim 24, further comprising determining a state of a breakpoint bit, and setting a breakpoint for an instruction if it is determined that said state of said breakpoint bit is set. See Glew, column 6, lines 24-38.

28. Referring to claim 26, Glew in view of Deng has taught the method of claim 24, further comprising running a debug process on a host device, and entering debug commands through a graphical user interface. See Deng, column 1, line 50, to column 2, line 9. Note that if a user is stepping through a program, then that user must enter a step command.

29. Referring to claim 27, Glew in view of Deng has taught the method of claim 24, wherein said at least three register bit fields comprise a run field, a single step field and a debug enable field. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and field 202 (BE1, BE2, BE3, or BE4), which is a run field.

30. Referring to claim 28, Glew in view of Deng has taught the method of claim 24, further comprising: determining a state of a single-step bit, entering commands for single-stepping through a set of instructions for a cycle if said state of said single-step bit is set. See Fig.16, field 206, and column 5, lines 29-33, of Deng.

31. Referring to claim 29, Glew in view of Deng has taught a method as described in claim 24.

a) Glew has further taught accessing internal states of each of a plurality of processors through said debug process, wherein accessing includes reading state values and overwriting state values. The examiner deems this to be inherent as this is the nature of debugging. During debugging, a user is able to view the state of the machine (and consequently, of each of the processors (each

superscalar processing element) and set the state of the machine in order to determine how the system is functioning.

b) Glew in view of Deng has not taught loading content of at least one register into an instruction memory and loading content of said instruction memory into the at least one register. However, Official Notice is taken that store and load instructions and their advantages are well known and accepted in the art. A store instruction, as is known, stores data in a register into an instruction memory (a memory which is accessed by an instruction), and a load instruction, as is known, retrieves data from instruction memory (a memory which is accessed by an instruction) and loads the data into a register. Such instructions allow a processor to read and write data to memory, thereby increasing its storage space. As a result, in order to increase storage flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew in view of Deng to include a load content of at least one register into an instruction memory (store instruction) and load content of said instruction memory into the at least one register (load instruction).

Response to Arguments

32. Applicant's arguments filed on August 13, 2007, have been fully considered but they are not persuasive.

33. First, regarding section II on page 10 of the remarks, it is noted that, to overcome the 35 U.S.C 101 rejection, applicant amended the specification to remove the disclosure regarding carrier waves and also expressed that there is no intent to claim the limitations of carrier waves. However, carrier waves were only one item deleted from the specification. Applicant

specifically deleted "biological electrical, mechanical systems, electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.)." Therefore, unless applicant states otherwise, it is assumed by the examiner that applicant has no intent to claim any of the deleted material.

34. Applicant argues the 35 U.S.C. 112, 1st paragraph rejections of claims 1, 18, and 24, on page 10 of the remarks, in substance that:

> "Applicant has amended the claims with the limitations of adding debug fields to the control status register field. it is well known in the art that fields can be added/attached to other fields within a register."

35. These arguments are not found persuasive for the following reasons:

a) The examiner asserts that merely asserting that attaching fields to a register is well-known does not make it so. It is still unclear to the examiner how to make an invention, without undue experimentation, that dynamically adds register fields to another field in a register unless the machine is reconfigurable (e.g. an FPGA), which applicant appears to have no support for. As is known, hardware is set at fabrication time. If an 8-bit control register was fabricated, then an 8-bit register is available during processing. Within this register are 8 predetermined bit fields. Therefore, it is not clear how fields can be added. One could call three of the fields a run field, a single step field, and a debug enable field, but again, they already exist. They are not added, or attached, but merely manipulated at run-time. The examiner will maintain this rejection until clarification is received. If applicant could provide documentation showing that this concept is well-known, it would be greatly appreciated.

36. Applicant argues the rejection of claims 1, 18, and 24, on page 13 of the remarks, in substance that:

> "Deng discloses a breakpoint control register (see Deng, Fig. 16, col. 9, lines 21-26). Deng, however, does not teach that 'said controller adds at least three debug register bit fields to at least one processor control status register field, wherein said at least three register bit fields comprise a run field, a single step field and a debug enable field.' That is, the processor control status register already exists with a predefined field structure. Applicant's claimed invention makes use of this preexisting structure and changes the field structure by adding at least three debug register fields. The processor control status register is not a breakpoint control register."

37. These arguments are not found persuasive for the following reasons:

a) The examiner's response to this argument is closely tied to the response above regarding the 112 rejection. Again, it is not clear how bits are added to a register. Applicant states that the field structure of a preexisting register is changed, but how? A register simply has a number of bit fields. To merely assign the aforementioned run, single step, and enable functionality to them does not constitute changing structure. It simply means that a bit is used to control each of the functionalities. If applicant is changing field structure in some other way, it is not evident from the specification, and further clarification is required. At this point, the examiner feels that Glew and Deng still teach the limitations of claim 1. Specifically, Deng has taught "adding" 3 bits to a register field to control specific functionality. They are added in the sense that they are multiple bits added to the remaining bits (field) of that register. Even if it happens at fabrication time, they are still added. They are also added in the sense that they are logically added to control functional behavior for three different components.

Conclusion

38. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJH
David J. Huisman
September 26, 2007